A Practical Multi~~level~~ -Index ~~Higher Order~~ Quasi-Monte Carlo Method for Simulating PDEs with Random Coefficients

Pieterjan Robbe
pieterjan.robbe@kuleuven.be

joint work with Dirk Nuyens and Stefan Vandewalle

April 6, 2016

KU Leuven - University of Leuven
Department of Computer Science
Celestijnenlaan 200A
B3001 Leuven, Belgium

$$\mathrm{MSE}(\mathcal{M}) = \mathbb{V}[\mathcal{M}] + \mathrm{Bias}(\mathcal{M}, G)^2$$

QMC

MIMC

?

Overview

The problem: PDEs with random coefficients

The methods: Multi-Index Monte Carlo...

... and Multi-Index Quasi-Monte Carlo

Conclusion

PDEs with random coefficients

Steady-state flow through porous media can be described by Darcy's law, taking the form of an elliptic PDE with random diffusion coefficient

$$-\nabla \cdot (k(\boldsymbol{x};\omega)\nabla p(\boldsymbol{x};\omega)) = f(\boldsymbol{x})$$

with $\boldsymbol{x} \in [0,1]^3$ and $\omega \in \Omega$

The diffusion coefficient is modelled as a lognormal random field with exponential covariance function

$$C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 \exp\left(\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p}{\lambda}\right)$$

This is a specific instance of the Matérn kernel

$$C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p}{\lambda/2\sqrt{\nu}}\right) K_\nu \left(\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p}{\lambda/2\sqrt{\nu}}\right)$$

PDEs with random coefficients

representation of uncertainty

We use the KL-expansion to take samples from the diffusion coefficient

$$k(\boldsymbol{x};\omega) = \bar{k} + \exp\left(\sum_{n=1}^{\infty} \sqrt{\theta_n} f_n(\boldsymbol{x}) \xi(\omega)\right) \tag{1}$$

with θ_n and f_n the eigenvalues and eigenfunctions of the covariance operator $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$

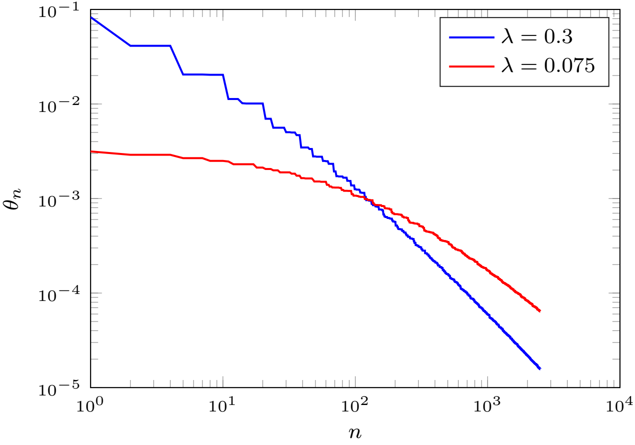
- ⋆ We use $p = 1$ (1-norm) in (1) because analytic expressions exist for eigenvalues and eigenfunctions
- ⋆ In practice, the infinite sum must be truncated after s terms

Also fast circulant embedding techniques are possible[1]

[1] see talk by Dirk Nuyens tomorrow (MS117)

PDEs with random coefficients

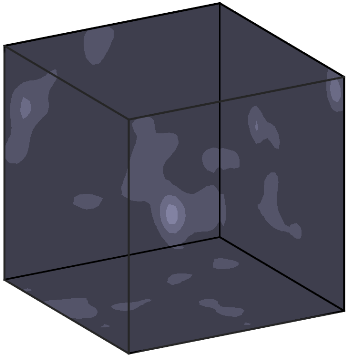
structure of the eigenvalues



Decay of the three-dimensional eigenvalues in the KL-expansion for $\lambda = 0.3$ and $\lambda = 0.075$. In both cases, the variance $\sigma^2 = 1$.

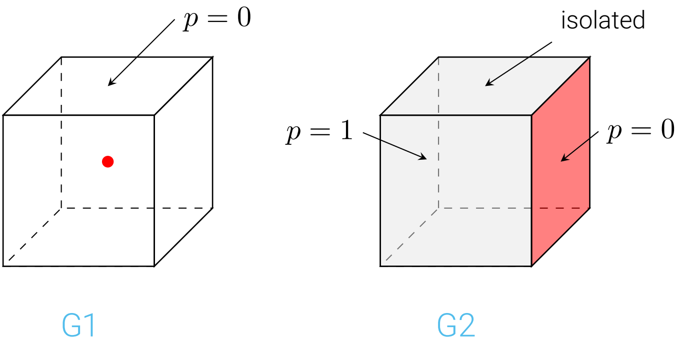
PDEs with random coefficients

a typical sample of the diffusion coefficient



PDEs with random coefficients

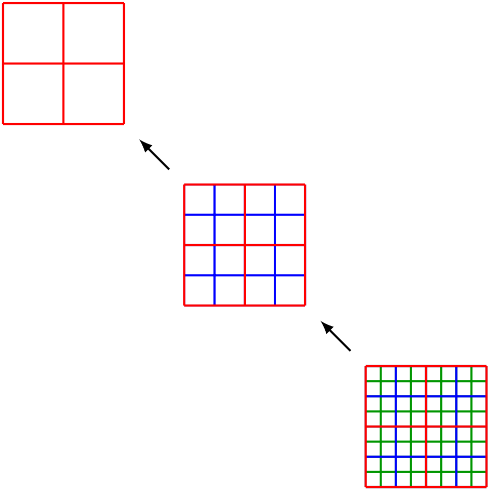
quantity of interest



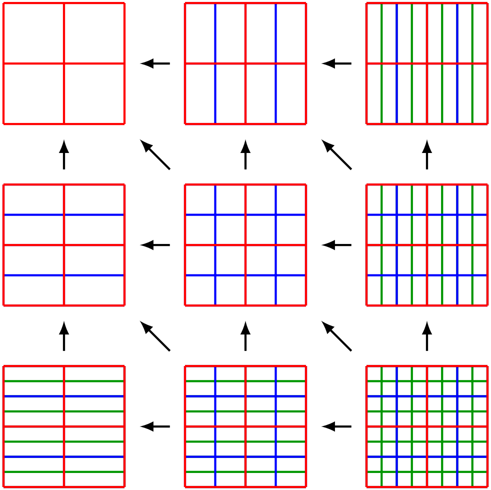
We are interested in the statistics of a quantity of interest G $:= \mathcal{G}(p(\boldsymbol{x}; \omega))$

In particular, what are $\mathbb{E}[\text{G1}]$ and $\mathbb{E}[\text{G2}]$?

Multilevel Monte Carlo



Multi-Index Monte Carlo



Multi-Index Monte Carlo

Define the difference operator along a single direction i, denoted Δ_i, by

$$\Delta_i = \begin{cases} G_{\boldsymbol{\ell}} - G_{\boldsymbol{\ell}-\boldsymbol{e}_i} & \text{if } \boldsymbol{e}_i \cdot \boldsymbol{\ell} > 0 \\ G_{\boldsymbol{\ell}} & \text{if } \boldsymbol{e}_i \cdot \boldsymbol{\ell} = 0 \end{cases},$$

where $\boldsymbol{e}_i$ is the unit vector in direction i

Next, we define the difference operator $\Delta = \prod_{i=1}^{d} \Delta_i$

Again, the expected value can be expressed as the telescoping sum

$$\mathbb{E}[G] = \sum_{\boldsymbol{\ell} \geq 0} \mathbb{E}[\Delta G_{\boldsymbol{\ell}}]$$

By choosing a suitable subset of all $\boldsymbol{\ell}$ we can reduce the bias of the estimator and avoid to take samples at $\boldsymbol{\ell} = (L, L, L)$

Multi-Index Monte Carlo

We consider three different types of index sets $\mathcal{I}$:

1. **Full Tensor (FT) index sets.** *

$$\mathcal{I}(L) = \left\{ \boldsymbol{\ell} \in \mathbb{N}^d : \ell_i \leq L \text{ for all } 1 \leq i \leq d \right\}$$

2. **Total Degree (TD) index sets.** *

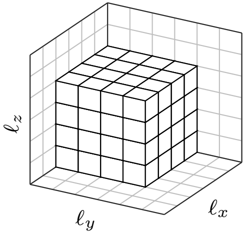
$$\mathcal{I}(L) = \left\{ \boldsymbol{\ell} \in \mathbb{N}^d : \sum_{i=1}^{d} \ell_i \leq L \right\}$$

3. **Hyperbolic Cross (HC) index sets.**

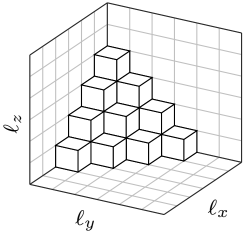
$$\mathcal{I}(L) = \left\{ \boldsymbol{\ell} \in \mathbb{N}^d : \prod_{i=1}^{d} (\ell_i + 1) \leq L \right\}$$

$*$ = introduced by Haji-Ali, Abdul-Lateef, Fabio Nobile, and Raúl Tempone. "Multi-index Monte Carlo: when sparsity meets sampling." *Numerische Mathematik* (2015): 1-40.

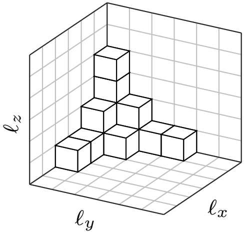
Multi-Index Monte Carlo



FT index set



TD index set



HC index set

Multi-Index Monte Carlo

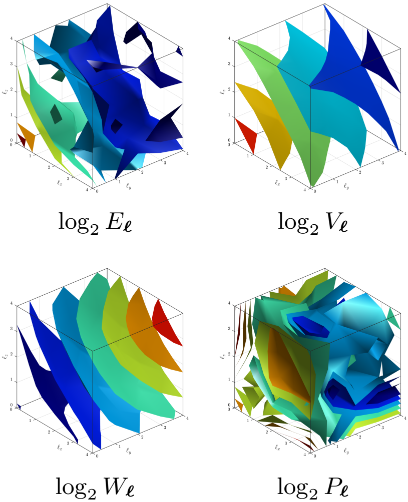
regularity conditions for G1

$$E_{\boldsymbol{\ell}} \leq C_E \prod_{i=1}^{d} 2^{-\ell_i \alpha_i}$$

$$V_{\boldsymbol{\ell}} \leq C_V \prod_{i=1}^{d} 2^{-\ell_i \beta_i}$$

$$W_{\boldsymbol{\ell}} \leq C_W \prod_{i=1}^{d} 2^{\ell_i \gamma_i}$$

with C_E, C_V, C_W, α_i, β_i and $\gamma_i > 0$ for $i = 1 \ldots d$.



Multi-Index Monte Carlo

performance for G1

Fitted orders of convergence for $E_{\boldsymbol{\ell}}$, $V_{\boldsymbol{\ell}}$ and $W_{\boldsymbol{\ell}}$:

direction	x	y	z
α_i	1.3443	1.3468	1.3425
β_i	4.0600	4.0329	4.0198
γ_i	1.1324	1.1776	1.1271

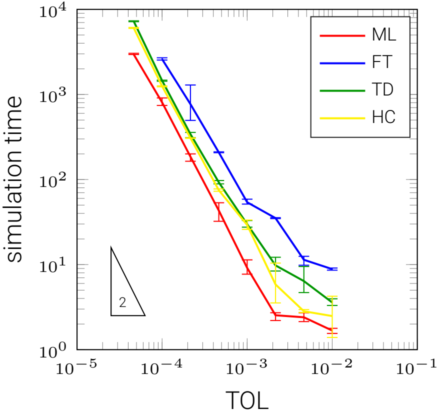
For all $i = 1 \ldots d$, $\alpha_i \approx \alpha$, $\beta_i \approx \beta$ and $\gamma_i \approx \gamma$ (isotropy)

Hence, we expect the optimal complexity rate

$$\text{Total Work} \lesssim \mathcal{O}(\epsilon^{-2}) \begin{cases} \text{if } \beta > d\gamma & \text{in the ML case} \\ \text{if } \beta > \gamma & \text{in the TD case} \end{cases}$$

Multi-Index Monte Carlo

performance for G1



Simulation details:

- ⋆ correlation length $\lambda = 0.3$, variance $\sigma^2 = 1$
- ⋆ $s = 250$ uncertainties
- ⋆ finest grid has $4 \cdot 2^5$ grid points in each dimension
- ⋆ the PDE is solved using a finite volume method
- ⋆ the sparse solver is CG with multigrid preconditioning as implemented in `hsl_mi20`

Multi-Index Monte Carlo

`MUMPS` vs `AMG`

We expect the optimal complexity rate

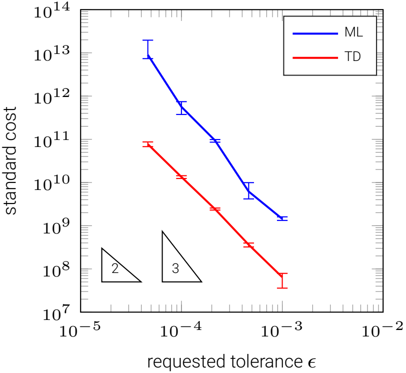
$$\text{Total Work} \lesssim \mathcal{O}(\epsilon^{-2}) \begin{cases} \text{if } \beta > d\gamma & \text{in the ML case} \\ \text{if } \beta > \gamma & \text{in the TD case} \end{cases}$$

So, MIMC performs better than MLMC when $\gamma < \beta < d\gamma$

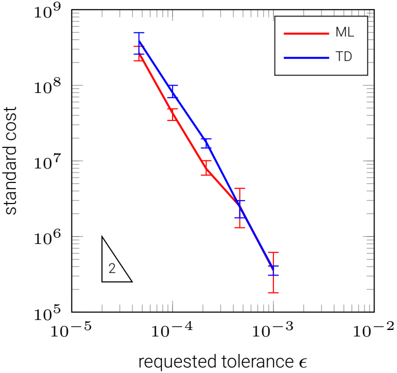
- ⋆ when using `MUMPS`: $\gamma \approx 2 \quad \Rightarrow \quad 2 < \beta < 6$ **true**
- ⋆ when using `AMG`: $\gamma \approx 1 \quad \Rightarrow \quad 1 < \beta < 3$ **false**

Multi-Index Monte Carlo

MUMPS vs AMG



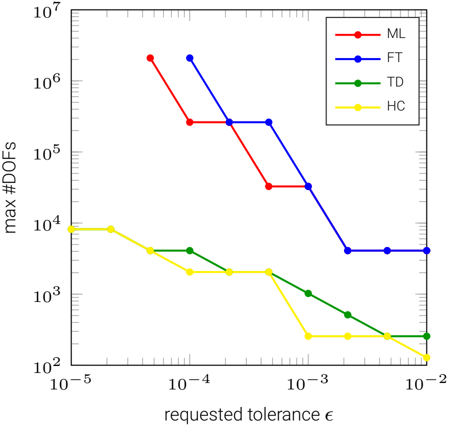
$\gamma \approx 2$



$\gamma \approx 1$

Multi-Index Monte Carlo

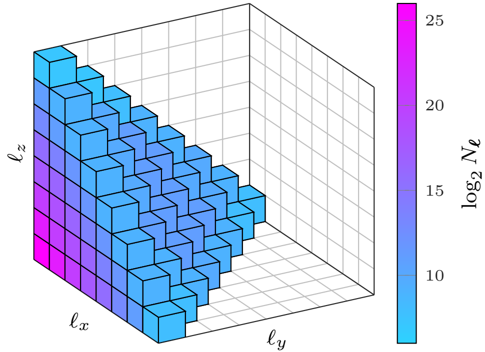
memory efficiency



Maximum number of degrees of freedom (DOF) versus requested tolerance for standard MLMC (ML) and MIMC with Full Tensor (FT), Total Degree (TD) and Hyperbolic Cross (HC) index sets.

Multi-Index Monte Carlo

total number of samples taken for MIMC with TD index sets, $\epsilon = \texttt{1e-4}$



Multi-Index Monte Carlo

performance for G2

Fitted orders of convergence for $E_{\boldsymbol{\ell}}$, $V_{\boldsymbol{\ell}}$ and $W_{\boldsymbol{\ell}}$:

direction	x	y	z
α_i	1.9683	1.0129	1.0137
β_i	4.4080	2.3344	2.3422
γ_i	1.2710	1.2527	1.2294

We find

$$\alpha_1 \approx 2\alpha_2 \approx 2\alpha_3 \approx 2\alpha$$

$$\beta_1 \approx 2\beta_2 \approx 2\beta_3 \approx 2\beta$$

$$\gamma_i \approx \gamma \text{ for all } i = 1 \dots d$$

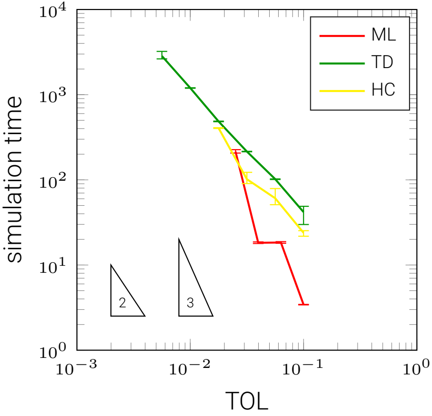
We obtain the following cost estimates:

$$\text{Total Work} \leq \mathcal{O}(\epsilon^{-3}) \text{ for MLMC, and}$$

$$\text{Total Work} \leq \mathcal{O}(\epsilon^{-2}) \text{ for MIMC (TD)}$$

Multi-Index Monte Carlo

performance for G2



Simulation details:

- ⋆ correlation length $\lambda = 0.075$, variance $\sigma^2 = 1$
- ⋆ $s = 3000$ uncertainties
- ⋆ finest grid has $4 \cdot 2^5$ grid points in each dimension
- ⋆ the PDE is solved using a finite volume method
- ⋆ the sparse solver is CG with multigrid preconditioning as implemented in `hsl_mi20`

Multi-Index Quasi-Monte Carlo

The basis of any multilevel method is the telescoping sum

$$\mathbb{E}[G_L] = \sum_{\boldsymbol{\ell} \in \mathcal{I}} \mathbb{E}[\Delta G_{\boldsymbol{\ell}}]$$

Let $\Delta \mathcal{Q}_{\boldsymbol{\ell}}$ be an unbiased estimator for $\Delta G_{\boldsymbol{\ell}}$, then the general multilevel estimator can be expressed as

$$\mathcal{M} = \sum_{\boldsymbol{\ell} \in \mathcal{I}} \Delta \mathcal{Q}_{\boldsymbol{\ell}}$$

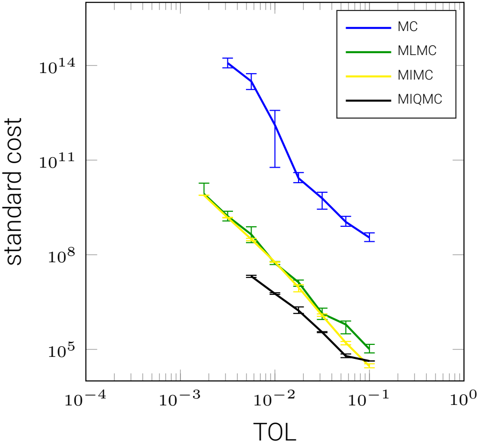
For example,

$$\mathcal{M} = \sum_{\boldsymbol{\ell} \in \mathcal{I}} \frac{1}{N_{\boldsymbol{\ell}}} \sum_{n=0}^{N_{\boldsymbol{\ell}}-1} \Delta G_{\boldsymbol{\ell}}(\xi_n)$$

with ξ_n $\begin{cases} \text{random numbers} & \text{for MIMC} \\ \text{randomly shifted rank-1 lattice points} & \text{for MIQMC} \end{cases}$

Multi-Index Quasi-Monte Carlo

first numerical results

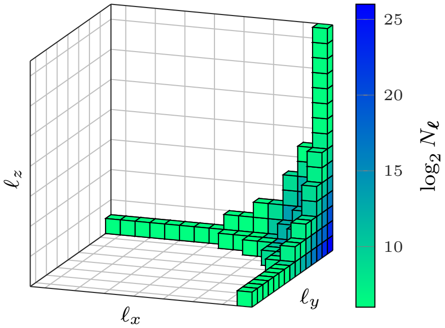


Comparison of Monte Carlo (MC), Multilevel Monte Carlo (MLMC), Multi-Index Monte Carlo (MIMC with TD index sets) and Multi-Index Quasi-Monte Carlo (MIQMC with TD index sets) for the first test problem G1

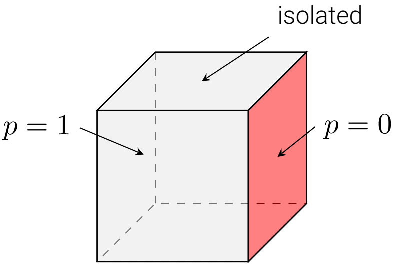
Questions & Remarks

Suggestions:

- ⋆ wait - can you exploit the directionality in the second test case?
- ⋆ but how does your algorithm for MIQMC simulation work?
- ⋆ and how do you choose the number of samples?
- ⋆ how did you choose the number of terms in the KL-expansion?

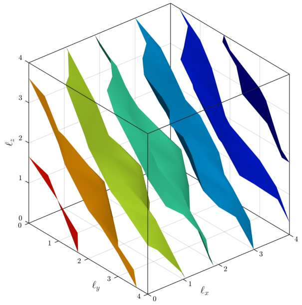


Exploiting the directionality for G2



G2

direction	x	y	z
α_i	1.9683	1.0129	1.0137
β_i	4.4080	2.3344	2.3422
γ_i	1.2710	1.2527	1.2294



Exploiting the directionality for G2

Question: can we exploit directionality of the problem?

Answer: yes, if we define

$$\mathcal{I}_{\boldsymbol{\delta}}(L) = \left\{ \boldsymbol{\ell} \in \mathbb{N}^d : \sum_{i=1}^{d} \delta_i \ell_i \leq L \right\} \qquad \text{for TD index sets}$$

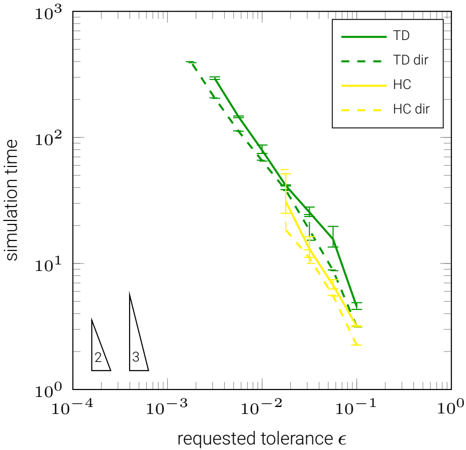
$$\mathcal{I}_{\boldsymbol{\delta}}(L) = \left\{ \boldsymbol{\ell} \in \mathbb{N}^d : \prod_{i=1}^{d} (\delta_i \ell_i + 1) \leq L \right\} \qquad \text{for HC index sets}$$

with $\delta = [2\ 1\ 1]$

Of course, this requires a priori knowledge of the problem!

back

Numerical Experiments



Performance of Multi-Index Monte Carlo with direction-aware index sets: run time versus requested tolerance for isotropic MIMC with Total Degree (TD) and Hyperbolic Cross (HC) index sets and direction-aware MIMC with TD and HC index sets.

MIQMC details

The mean-square error (MSE) of $\mathcal{M}$ can be expanded as

$$\begin{aligned}\mathrm{MSE}(\mathcal{M}) &= \mathbb{E}[(\mathcal{M} - \mathbb{E}[\mathcal{M}])^2] + (\mathbb{E}[\mathcal{M}] - G)^2 \\ &= \mathbb{V}[\mathcal{M}] + \mathrm{Bias}(\mathcal{M}, G)^2\end{aligned}$$

We bound both terms as

$$\mathrm{Bias}(\mathcal{M}, G) \leq (1-\theta)\epsilon, \text{ and} \qquad \text{(bias constraint)}$$

$$\mathrm{prob}[|\mathcal{M} - \mathbb{E}[\mathcal{M}]| \leq \theta\epsilon] \geq 1 - \nu, \qquad \text{(statistical constraint)}$$

where ϵ is a tolerance, θ the error splitting and ν a failure probability

Note that, by normality of the estimator, we can rewrite the statistical constraint as

$$\mathbb{V}[\mathcal{M}] \leq (\theta \mathrm{TOL})^2 \quad \text{with} \quad \mathrm{TOL} := \frac{\epsilon}{\Phi^{-1}(1 - \nu/2)}$$

Optimal number of samples

The optimal number of samples N_ℓ is the solution of

$$\begin{aligned} &\min_{N_{\boldsymbol{\ell}}} \text{Total Work} = \sum_{\boldsymbol{\ell} \in \mathcal{I}} N_{\boldsymbol{\ell}} W_{\boldsymbol{\ell}} \\ &\text{s.t.} \quad \mathbb{V}[\mathcal{M}] \leq \text{TOL}^2 \end{aligned}$$

For MIMC, we have that

$$\mathbb{V}[\mathcal{M}] = \sum_{\boldsymbol{\ell} \in \mathcal{I}} \frac{V_\ell}{N_\ell}$$

For MIQMC, we find

$$\mathbb{V}[\mathcal{M}] \lesssim \sum_{\boldsymbol{\ell} \in \mathcal{I}} \frac{V_\ell}{K N_\ell^{-2}}$$

Optimal number of samples

The MIMC estimator is

$$\mathcal{M} := \sum_{\boldsymbol{\ell}\in\mathcal{I}} \mathcal{Q}_{s,N_\ell}^{K}(\Delta G_\ell) = \sum_{\boldsymbol{\ell}\in\mathcal{I}} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_\ell} \sum_{n=0}^{N_\ell - 1} \Delta G_\ell \left(\boldsymbol{t}_n + \boldsymbol{\Delta}_{k,\ell}\right)$$

For a rank-1 lattice rule,

$$\boldsymbol{t}_n = \left\{ \frac{n\boldsymbol{z}}{N} \right\}, \quad n = 0, \ldots, N-1,$$

where $\boldsymbol{z} \in \mathbb{Z}^s$ is a generating vector, and $\{\ \cdot\ \}$ denotes the fractional part

$\boldsymbol{\Delta}_{k,l} \sim \mathcal{U}(0,1)$ is a random shift

Optimal number of samples

$$
\begin{aligned}
\mathbb{V}_{\boldsymbol{\Delta}}[\mathcal{M}_{\text{QMC}}] &= \mathbb{V}_{\boldsymbol{\Delta}}\left[\sum_{\boldsymbol{\ell}\in\mathcal{I}} \mathcal{Q}^{K}_{s,N_\ell}(\Delta G_\ell)\right] \\
&= \sum_{\boldsymbol{\ell}\in\mathcal{I}} \mathbb{V}_{\boldsymbol{\Delta}}\left[\mathcal{Q}^{K}_{s,N_\ell}(\Delta G_\ell)\right] \\
&= \sum_{\boldsymbol{\ell}\in\mathcal{I}} \mathbb{E}_{\boldsymbol{\Delta}}\left[|\mathcal{Q}^{K}_{s,N_\ell}(\Delta G_\ell) - \mathbb{E}_{\boldsymbol{\Delta}}[\Delta G_\ell]|^2\right] \\
&\leq \sum_{\boldsymbol{\ell}\in\mathcal{I}} \frac{1}{K} e^{\mathrm{sh}}_{\mathrm{wor}}(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)^2 \|\Delta G_\ell\|^2_{\mathscr{H}} \\
&\leq \sum_{\boldsymbol{\ell}\in\mathcal{I}} \frac{1}{K} \left(\frac{\mathcal{C}_{s,\lambda}}{N_\ell^\lambda}\right)^2 \|\Delta G_\ell\|^2_{\mathscr{H}}
\end{aligned}
$$

where $\mathscr{H}$ is the weighted and unanchored Sobolev space of functions on $[0,1]^s$ with square integrable mixed first-order derivatives, and we assume that a similar bound for the worst-case error is valid for integration in $\mathbb{R}^s$

We assume a QMC method that converges as $\mathcal{O}(1/N^\lambda)$

Optimal number of samples

Hence, we find

$$N_\ell \eqsim \sqrt[2\lambda+1]{\frac{2\lambda \mathcal{C}_{s,\lambda}^2 \|\Delta G_\ell\|_{\mathscr{H}}^2}{K^2 W_\ell}}$$

Because this analysis is also valid for the usual MIMC setting, we propose $\mathcal{C}_{s,\lambda}^2 \|\Delta G_\ell\|_{\mathscr{H}}^2 \lesssim V_\ell$

We find the optimal number of samples as

$$N_\ell \geq \sqrt[2\lambda]{(\theta \mathrm{TOL}_{\epsilon,\nu})^{-2} \frac{1}{K} \left(\frac{V_\ell}{W_\ell}\right)^{\frac{2\lambda}{2\lambda+1}} \sum_{m \in \mathcal{I}} \sqrt[2\lambda+1]{V_m W_m^{2\lambda}}}$$

back

MIQMC algorithm

```
begin
    L := −1; θ := 0.5; I_0 = ∅; converged := false;
    repeat
        L ← L + 1;
        take N* samples at each index ℓ ∈ I(L)\I(L − 1);
        compute sample variance and bias at each level ℓ;
        if bias < ε/2 then
            θ ← 1−bias/ε;
        end
        compute optimal number of samples at each ℓ;
        update samples at each ℓ ∈ I(L);
        while variance of estimator > θ · ε do
            double number of samples where it is most beneficial;
        end
        if L > 2 then
            recompute bias and check for convergence;
        end
    until converged = true;
end
```